

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号
特表2001-520775
(P2001-520775A)

(43) 公表日 平成13年10月30日 (2001. 10. 30)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 F 7/72		G 0 6 F 7/72	
G 0 9 C 1/00	6 5 0	G 0 9 C 1/00	6 5 0 A 6 5 0 Z

審査請求 未請求 予備審査請求 有 (全 37 頁)

(21) 出願番号 特願平10-544618
(86) (22) 出願日 平成10年4月20日 (1998. 4. 20)
(85) 翻訳文提出日 平成11年10月18日 (1999. 10. 18)
(86) 国際出願番号 P C T / C A 9 8 / 0 0 4 6 7
(87) 国際公開番号 W O 9 8 / 4 8 3 4 5
(87) 国際公開日 平成10年10月29日 (1998. 10. 29)
(31) 優先権主張番号 9 7 0 7 8 6 1. 2
(32) 優先日 平成9年4月18日 (1997. 4. 18)
(33) 優先権主張国 イギリス (G B)

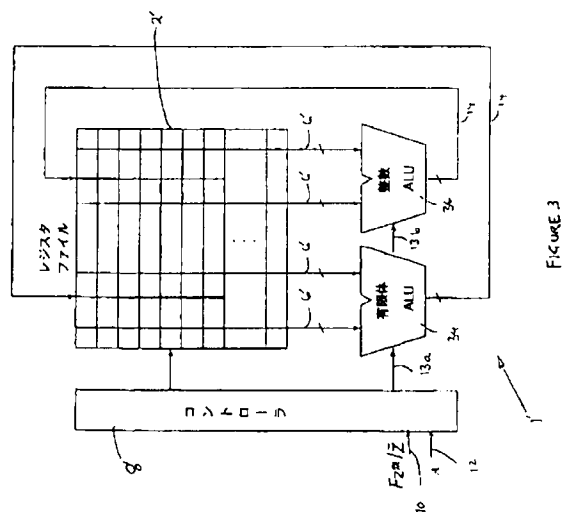
(71) 出願人 サーティカム コーポレイション
カナダ, オンタリオ州 エル4ダブリュー
5エル1, ミシソーガ フォース フロ
ア, エクスプローラー ドライブ 5520
(72) 発明者 バンストーン, スコット, エイ.
カナダ エヌ2ティー 2エイチ4 オン
タリオ ワーテルロー サンドブロック
コート 539
(72) 発明者 ランベルト, ロバート, ジェイ.
カナダ エヌ2エル 3ビー8 オンタリ
オ ワーテルロー ナンバー 212 シー
グラム ドライブ 106
(74) 代理人 弁理士 谷 義一 (外2名)

最終頁に続く

(54) 【発明の名称】 算術プロセッサ

(57) 【要約】

この開示は、有限体演算やモジュラ整数演算など一群の関連する算術演算をそれぞれパフォームする複数の算術回路を有するALUを含むことを特徴とする算術プロセッサを提供するものである。ALUは、オペランドデータを受信するオペランド入力データバスと、算術演算の結果を戻す結果データ出力バスとを有する。レジスタファイルはオペランドデータバスと結果データバスに結合されている。レジスタファイルは複数の算術回路によって共用されている。さらに、コントローラがALUおよびレジスタファイルに結合されていて、このコントローラは、算術演算を要求するモード制御信号に応答して、複数の算術回路の1つを選択し、レジスタファイルとALUとの間でデータアクセスを制御し、それによりレジスタファイルが算術回路によって共用されるようにする。



【特許請求の範囲】

1. (a)有限体算術演算をパフォームする有限体算術回路とモジュラ整数算術演算をパフォームするモジュラ整数算術回路を有する論理演算装置であって、オペランドデータを受信するオペランド入力データバスと前記算術演算の結果を戻す結果データ出力バスとを有する論理演算装置と、

(b) 前記オペランドデータバスおよび前記結果データバスに結合したレジスタファイルと、

(c) 前記ALUおよび前記レジスタファイルに結合したコントローラであって、モード制御信号に応答して前記有限体演算または前記整数算術演算のいずれかを選択し、前記レジスタファイルと前記ALUの間でのデータアクセスを制御して、前記レジスタファイルが前記有限体算術回路および整数算術回路の両方によって共用されるようにしたコントローラとを備えたことを特徴とする算術プロセッサ。

2. 請求項1において、

前記レジスタファイルは汎用レジスタを含み、

前記ALUは前記オペランドバスのデータビット幅より広い処理ビット幅を有すること
を特徴とする算術プロセッサ。

3. 請求項1において、前記コントローラは、前記論理演算装置により選択された算術演算を制御する命令をプログラムしたことを特徴とする算術プロセッサ。

4. 請求項1において、前記オペランドバスは、前記ALUの処理ビット幅および前記結果データバスのビット幅と同じビット幅を有することを特徴とする算術プロセッサ。

5. 請求項4において、前記オペランドデータバスは、第1および第2のオペラ

ンドをそれぞれ前記ALUと結合する第1および第2のオペランドバスを含むことを特徴とする算術プロセッサ。

6. 請求項5において、前記汎用レジスタは、

前記コントローラによって個別にアドレスすることができ、

前記ALUの前記処理ビット幅より大きな体サイズについて前記ALUで計算するために、複数のレジスタ中のデータを組み合わせる

ことができることを特徴とする算術プロセッサ。

7. 請求項1において、前記コントローラが体サイズの制御に応答して、前記ALUが種々の体サイズに作用することができることを特徴とする算術プロセッサ

。

8. 請求項1において、

前記ALUは、

前記算術演算で利用するオペランドを前記レジスタファイルから受信する複数の特殊レジスタと、

前記特殊レジスタの1つ以上のビットを結合する組合せおよびロジック回路要素を有する複数のサブALUと、

前記コントローラから受信した制御情報に応答する順序づけコントローラとを含み、

前記順序づけコントローラ、ならびにその中のカウンタおよび検出回路は、前記特殊レジスタおよび前記複数のサブALUに結合され、算術演算中の一連のステップがパフォームされるようにその演算を制御する

ことを特徴とする算術プロセッサ。

9. 請求項8において、前記ALUは、有限体の乗算と、二乗と、加算と、減算と、反転の前記算術演算をパフォームすることを特徴とする算術プロセッサ。

10. 請求項8において、前記サブALUは、XORと、シフトと、シフトXO

Rと、加算と、減算論理演算をパフォームすることを特徴とする算術プロセッサ

。

11. 請求項1において、前記有限体算術回路は、

第1および第2のオペランドビットベクトルをそれぞれ受信するAレジスタおよびBレジスタと、モジュラスビットベクトルを受信するMレジスタと、前記オペランドの有限体の積を含むアキュムレータとを含む複数の特殊レジスタを有する有限体乗算器回路と、

前記AおよびBレジスタの各セルから前記アキュムレータのセルへの接続を確立するロジック回路と、

前記レジスタおよび前記ロジック回路に動作可能に接続され、前記有限体の積を得る一連のステップを実施する順序づけコントローラとを含むことを特徴とする算術プロセッサ。

12. 請求項11において、前記一連のステップは、

前記Aレジスタの内容と前記Bレジスタの連続ビットの部分積を計算するステップと、

前記部分積を前記アキュムレータにストアするステップと、

前記部分積のビットを試験するステップと、

前記テストしたビットがセットされた場合に前記部分積を前記モジュラスで簡約するステップと、

前記Bレジスタの連続ビットについて前記ステップを繰り返すステップとを備えたことを特徴とする算術プロセッサ。

13. 請求項12において、

前記左そろえのオペランドベクトルを前記Aレジスタおよび前記Bレジスタにそれぞれストアするステップを含み、

前記テストビットは前記レジスタの前記左端のビットから得られることを特徴とする算術プロセッサ。

14. 請求項12において、前記Bレジスタがシフトレジスタであることを特徴とする算術プロセッサ。

15. 請求項14において、前記ロジック回路は、

レジスタセル A_i およびアキュムレータ C_i から得られる入力を有し、レジスタBのセル B_{N-1} から得られた第1の加算制御信号に応答して第1の加算出力信号を生成する第1の制御可能加算器と、

モジュラスレジスタセル M_i および前記加算出力信号から得られる入力を有し、前記アキュムレータのセル C_{N-1} から得られた第2の加算制御信号に応答してアキュムレータセル C_i に結合される出力を生成する第2の制御可能加算器と

をそれぞれ含む、それぞれのレジスタセルに結合した複数の制御可能加算装置を有することを特徴とする算術プロセッサ。

16. 請求項15において、有限体加算器回路を含むことを特徴とする算術プロセッサ。

17. 請求項16において、前記有限体加算器は、レジスタBの前記セルB_iから得られた入力を前記第1の加算器それぞれに結合する手段と、前記第2の加算器の前記出力を前記セルC_iに結合する手段とを含み、

前記順序づけコントローラが有限体加算制御信号に応答し、それにより前記有限体加算演算が1クロックサイクルでパフォームされることを特徴とする算術プロセッサ。

18. 請求項1において、前記有限体算術回路は有限体反転回路を含むことを特徴とする算術プロセッサ。

19. 請求項18において、前記有限体反転回路は、

第1および第2のオペランドビットベクトルをそれぞれ受信するAレジスタおよびBレジスタと、モジュラスビットベクトルを受信するMレジスタと、前記オ

ペランドの有限体の積を含むアキュムレータとを含む複数の特殊レジスタを備えたことを特徴とする算術プロセッサ。

20. 請求項1において、前記ALUは、

有限体乗算器回路と、

有限体反転回路と、

複数の特殊レジスタと、

前記特殊レジスタの各セルの間で接続を確立するロジック回路と、

前記レジスタおよび前記ロジック回路にアクション可能に接続され、有限体の積または有限体の反転を計算するための一連のステップを実施し、それにより前記特殊レジスタが前記有限体乗算器および前記有限体反転回路によって共用されるようにする、順序づけコントローラとを備えたことを特徴とする算術プロセッサ。

21. 請求項20において、前記有限体反転回路は、拡張したユークリッドの互

除法を実施することを特徴とする算術プロセッサ。

22. 請求項11において、整数算術乗算回路を含むことを特徴とする算術プロセッサ。

23. 請求項12において、前記整数算術乗算は、前記モード選択信号に応答して前記mレジスタに桁上げをロードすることによって実装されることを特徴とする算術プロセッサ。

24. 請求項1において、暗号化システムで使用されることを特徴とする算術プロセッサ。

25. (a)一群の関連する算術演算をそれぞれパフォームする複数の算術回路を有するALUであって、オペランドデータを受信するオペランド入力データバス

と、前記算術演算の結果を戻す結果データ出力バスとを有するALUと、

(b) 前記オペランドデータバスおよび前記結果データバスに結合したレジスタファイルと、

(c) 前記ALUおよび前記レジスタファイルに結合したコントローラであって、算術演算を要求するモード制御信号に応答して前記複数の算術回路の1つを選択し、かつ前記レジスタファイルと前記ALUの間でデータアクセスを制御し、それにより前記レジスタファイルが前記算術回路によって共用されるようにするコントローラと

を備えたことを特徴とする算術プロセッサ。

26. 請求項25において、前記算術回路は、有限体算術回路およびモジュラ整数算術回路であることを特徴とする算術プロセッサ。

【発明の詳細な説明】

算術プロセッサ

本発明は、有限体および整数の算術をパフォームする方法および装置に関する

。

発明の背景

有限体 (finite field) に対する EC (Elliptic Curve) 暗号法では、加算と、乗算と、二乗と、反転 (inversion) の算術演算が必要となる。さらに、体 (field) の標数が 2 でない場合には、減算も必要になる。例えば符号定数の計算では、モジュラ算術演算も必要になるが、このような演算は有限体の演算ほど必要にならない。例えば EC 暗号法では、モジュラおよび有限体演算、加算、減算、乗算、反転の完全な補集合 (full complement) が必要になる。

暗号法のための体のサイズは比較的大きくなる傾向があり、算術演算を許容時間内で実行するために高速の専用プロセッサが必要になる。したがって、高速モジュラ算術プロセッサか、 F_{2^n} の算術演算をパフォームする専用プロセッサのいずれかが、多数インプリメントされている。特殊目的または専用プロセッサを使用することは、当技術分野では、周知のことである。こうしたプロセッサは一般にコプロセッサと呼ばれ、通常は、ホスト計算システムで利用されており、従って、命令および制御はメインプロセッサからコプロセッサに提供されている。

なかでも RSA が暗号化システムとして慣用されていたが、優秀かつよりセキュアな EC 暗号法が登場したため、モジュラ冪法 (modular exponentiation) 専用のプロセッサの必要性は、薄らいで来ている。ユーザは RSA 暗号法から EC 暗号法に移行しているものの、性能およびコストをほとんどあるいは全く犠牲にすることなくこれらの両方の演算をサポートする算術プロセッサに対するニーズがある。

発明の概要

本発明の第 1 の目的は、有限体算術と整数算術を組み合わせ、EC 暗号法に必要な演算と、例えば RSA 暗号法で必要なモジュラ冪法とを提供するプロセッサ

を提供することにある。

本発明の第2の目的は、異なる体またはレジスタサイズにスケーリングすることのできる算術プロセッサ設計を提供することにある。

本発明の第3の目的は、異なる体サイズで 사용할 ことができる算術プロセッサを提供することにある。

本発明の第4の目的は、マルチシーケンス中の複数のステップを同時にパフォーマンスすることによってマルチシーケンス演算実行を高速化するためにスケーリングすることのできる算術プロセッサを提供することにある。

本発明によれば、

(a) 一群の関連する算術演算をそれぞれ実行する複数の算術回路を有する論理演算装置であって、オペランドデータを受信するオペランド入力データバスと、前記算術演算の結果を戻す結果データ出力バスとを有する論理演算装置と、

(b) 前記オペランドデータバスおよび前記結果データバスに結合されたレジスタファイルと、

(c) 前記ALUおよび前記レジスタファイルに結合された制御装置であって、算術演算を要求するモード制御信号に 応答して前記複数の算術回路の1つを選択し、かつ前記レジスタファイルと前記ALUの間でデータアクセスを制御することにより、前記レジスタファイルが前記算術回路によって共用されるようにする制御装置と

を含む算術プロセッサが提供される。

本発明の別の実施形態によれば、有限体回路および整数算術回路を含み、かつ汎用レジスタおよび専用レジスタを備えたプロセッサが提供される。

本発明の別の実施形態によれば、有限体算術および整数算術を両方とも実行し、専用レジスタおよび汎用レジスタの両方ならびに算術回路を共用する算術プロセッサが提供される。この目的のために、多項式基底が整数の標準的な基数累乗基底 (standard radix-power basis) と同様であるので、有限体ハードウェアについて多項式基底を想定する。

図面の簡単な説明

以下、本発明の実施形態を、添付図面を参照し例を挙げて説明する。

図1は有限体算術および整数算術をパフォームする算術プロセッサアーキテクチャのブロック図である。

図2は図1に示すALU (arithmetic logicunit) の概略ブロック図である。

図3は有限体算術および整数算術をパフォームする算術プロセッサアーキテクチャの代替実施形態のブロック図である。

図4は図3に示すALUの概略ブロック図である。

図5(a)、(b)、および(c)は図2に示すALUのビットスライスの実施形態のブロック図である。

図6は図5に示すビットスライスの有限体乗算器の回路図である。

図7は算術インバータのブロック図である。

図8は組み合わせた有限体／整数乗算器の回路図である。

図9は図1のマルチビットALUの実施形態を示す概略ブロック図である。

図10は図9のマルチビット有限体乗算器の回路図である。

好ましい実施形態の詳細な説明

図1を説明する。算術プロセッサの一実施形態は一般的に参照番号1で示してある。当然のことであるが、この算術プロセッサは総合計算システム中の汎用プロセッサとともに使用することができ、データはこの計算システムと算術プロセッサの間で交換される。算術プロセッサには、レジスタファイルと呼ばれる一群の汎用レジスタ(GP)2が含まれている。レジスタファイルはECの点加算(point addition)、点倍加(point doubling)などのための中間記憶域として使用することができるものである。一群の汎用レジスタ2はデータ入力またはオペランドバス6を介してALU (arithmetic logicunit) 4と通信を行なっている。ALU 4にはシェアード(shared)有限体および整数算術回路が含まれている。ALU 4による計算の結果をレジスタファイル2に書き込むため、データ出力または結果バス14がALU 4とレジスタファイル2の間に設けてある。

ALU 4による計算オペレーションは、算術プロセッサ1のコントローラ8に

駐在するマイクロプログラム化命令により制御されている。モード選択コントロール10は有限体計算またはモジュラ整数計算を選択するために用意してある。

体サイズ制御12も、ALU4を初期設定して、種々のオペランドベクトルサイズに適応させるために用意してある。そこで、コントローラ8はなにかんづく次のタスク、すなわち、適正な算術モードおよび演算をALU4に提供するタスクと、レジスタファイル2とALU4の間のデータアクセスをコーディネートするタスクと、使用される適正な体のサイズをALU4に提供するタスクとをパフォームする。

汎用レジスタは、少なくとも予想可能な最大の $F_{2m}EC$ 暗号システムをハンドルするだけのビット幅を有するように選択される。これら汎用レジスタは整数モジュラ算術に必要なビット長をサポートするために、組み合わせることができる。例えば、レジスタファイル2の単一レジスタのビット幅が512ビット幅である場合に、単一の2048ビットRSA量の記憶域を提供するため、4つのレジスタを使用することができる。これら汎用レジスタには、データのブロックがロードされ、例えば、2048ビットの計算をブロック単位で行い、ついで、再組み立てして、全幅結果(full width result)を得ることができる。典型的には、算術プロセッサ1は既存のホストコンピュータシステムで利用され、コントローラ8はこのホストコンピュータシステムから制御信号を受信し、適正なホストバスインタフェースを介して、ホストデータバスにデータを通信する。このようなインタフェースの詳細は当業者にとって周知のことであり、説明は省略する。

図2を説明する。ALU4には、幾つかの特殊レジスタ16と、複数のサブALU18と、出力データバス30と、コントローラ20とが含まれている。複数のサブALU18には組み合わせロジックおよび算術回路が含まれていて、組み合わせロジックおよび算術回路は、特殊レジスタからデータバス28を介して各サブALUに入力された1つ以上のビットをオペレートする。出力データバス30はサブALU18と特殊レジスタとの間に設けてある。コントローラ20は、なにかんづく、次のタスク、すなわち、計算オペレーション中の各ステップを通してALU4を順序づけるタスクと、特殊レジスタ16からの制御ビットを監視するタスクと、使用してある体のサイズを決定するためにカウンタを制御レジスタ

22に実装するとともに、プロセッサハードウェアを設計し直さずに、プロセッ

サ1が異なる体サイズに対して使用することができる機構を実装するタスクとをパフォームする。これらの機能を提供するため、特殊レジスタ16の制御ビット26は制御ビット入力24としてコントローラ20に供給される。特殊レジスタ16は、全て、個別にアドレス可能になっている。コントローラ20はレジスタファイルから入力バス6を介してサブALU18または特殊レジスタ16に入力されたデータも制御する。これらサブALUは、単一のビットにオペレートすることができ、複数のビットに一度にオペレートすることができる。これらのコンポーネントは後程より詳細に記述する。

図3を説明する。算術プロセッサの代替例は参照番号1'で示してある。本実施形態では、別個の有限体装置34および整数モジュラ算術装置36を提供する。このプロセッサは、レジスタファイル2'と、データ入力バス6'と、データ出力バス14'と、コントローラ8'も含むが、制御13aおよび13bがそれぞれコントローラ8'から各ALU34および36に提供される。

図4を説明する。図4は図3のALU34および36をより詳細に示す。ALU34および36には、それぞれ、特殊レジスタ16'aおよび16'bと、コントローラ20'aおよび20'bとが含まれている。ALU34および36には、それぞれ、サブALU18'aおよび18'bが含まれている。したがって、この実施形態では、特殊レジスタ16'aおよび16'bと、算術および制御回路は、当然、共有されない。サブALU18'aのうちの1つ以上のサブALU18'aは、協働して、シフト左/右とXORシフトの機能を実行し、サブALU18'bのうちの1つ以上のサブALU18'bは、協働し、任意選択で、桁上げ保存技術または桁上げ伝搬 (carry propagation) を使用して、整数加算および整数減算の機能を実行する。

図2を説明する。サブALU18は、特殊レジスタ16から供給されたオペランドに対して、次の論理機能、すなわち、XORと、シフト左/右と、XORシフトと、整数加算と、整数減算を実行する。これらの機能は、1つのサブALU18か、マルチプル・サブALUに含めることができる。マルチプル・サブALU18を設けることにより、当該プロセッサは複数の演算 (例えば、有限体反転)

を同時にパフォームすることができる。

図5を説明する。図5は図2のALU4のビットスライス41を詳細に示す。次の考察では、ビットスライス41と関係付けをしたロジック回路と関連して、各特殊レジスタのセルを相互接続する、と言う。ビットスライスに含まれたロジック回路は、一般的に、図2に示すようなサブALU18のうちの1つで表される。ビットスライスの構成は、Nビットレジスタに対しては、N回繰り返えすことができる。さらに、明確にするため、Nをレジスタ内のセル数と定義し、レジスタ内の個別のセルを例えば A_i という。ここで、 $0 \leq i \leq N-1$ であり、 A_{N-1} は特殊レジスタの最も右にあるセルである。レジスタの内容は小文字で参照され、例えば、長さnのビットベクトルAは、 a_0 をLSBとして、ビットに a_0, \dots, a_{n-1} と番号が付けられることになる。ここで、特殊レジスタには特定の名前が付けられているが、これらのレジスタは、後程説明するが、実行されている算術演算に依存して異なる機能をとることができる、ことに留意されたい。

図5の特殊レジスタ16に含まれるレジスタとしては、乗算演算中に、例えば、被乗数および乗数を個々に保持するための一対のオペランドレジスタA42およびB44と、累算器レジスタC46と、モジュラスレジスタM48と、桁上げ拡張 (carry extension) レジスタC^{ext}50 (整数算術で使用される) とがある。

これらのレジスタは、その中にロードされたビットベクトルの個々の2進数を保持するため、N個のセルを有する。これらのレジスタはシフトレジスタであるのが好ましい。図2に示すサブALU18は、後程説明するが、図5のブロック52の回路により実装することができる。

乗算

ALU4のオペレーションは、有限体乗算のような具体的な算術演算を参照することにより最も良く理解することができる。ここで、2つの元aおよびbの積Cを考察することにする。ここで、aおよびbはビットベクトルであり、bは多項式表現で $b = (b_0, \dots, b_{n-1})$ の形態となり、aは多項式表現で $a = (a_0, \dots, a_{n-1})$ の形態となる。モジュラスビットベクトルmは、 $m = (m_0, \dots, m_n)$ の形態を有する。モジュラスレジスタは、モジュラスを表すのに必要なビット数

より

1ビット多い、ことに留意されたい。あるいはまた、最上位ビット m_n が1であるので、この最上位ビットを暗黙に定義することができ、 m を (m_0, \dots, m_{n-1}) で表すこともできる。 F_{2^n} において、乗算は、次のような疑似コードにより明確に記述される一連のステップとして実装することができる。

$C = 0 \quad \{C_{-1} = 0\}$

For i from $n-1$ to 0 do

For j from $n-1$ to 0 do $\{C_j = C_{j-1} + b_j a_j + c_{n-1} m_j\}$

この乗算を実行する際には、MSB (most significant bit) からLSB (least significant bit) の順に、被乗数と乗数の b_j の各ビットとの部分積を形成する。その前の部分積のMSBがセットされた場合には、部分積はモジュラスによって簡約 (reduce) される。

乗算の実装は、 $1 \times N$ 乗算器を逐次使用することによって行なうことができ、この場合、上記疑似コードの内側の「for」ループはパラレルに実行される。各セルがそれぞれ2進数 m_j の1つを含むように、モジュラスレジスタ M には、MSB m_n を剥ぎ取ったモジュラスビットベクトル m がロードされる。図示の実装では、ビット m_j は、ベクトルのMSBを最も左側のビットとして、左から右に配列されている。すなわち、セル M_{n-1} はビット m_{n-1} を含む。 $N \neq n$ である場合、スティルビット (still bit) M_{n-1} は M_{N-1} にストアされる、すなわち、データは左寄せされる。各セルが個々に2進数 a_j または b_j の1つを含むように、シフトレジスタ A および B には、有限体元 (finite field element) ビットベクトル a および b がそれぞれロードされる。有限体元 a および b は、左寄せされ、各レジスタにストアされ、乗数レジスタ b のMSBが常に左境界セルのビット、すなわち $(a_{n-1}, a_{n-2}, \dots, a_0)$ および $(b_{n-1}, b_{n-2}, \dots, b_0)$ で利用可能になっている。ベクトル a および b の長さがレジスタの長さより短い場合には、残りのセルには0がパディングされる。以上、図2に示すコントローラ20によって一般的に実行される。逐次乗算 (被乗数を逐次小さくするなど) の他の構成も可能であるが、そのような構成では、体のサイズに柔軟性を持たせることができ

ないし、同様に、制御ビット位置を固定することができない。この乗算アルゴリズムを対応して変化させれば、LSBからMSBへのビット順序づけも可能である。

ここでは、ALU4のビットスライス41は、有限体において乗算を実装するために、記載されている。ビットスライス41は第1および第2のコントローラブル加算器54および56を含み、第1および第2のコントローラブル加算器54および56は、それぞれ、XOR機能を有する。レジスタBの最上位のセル B_{N-1} は、加算制御信号 b_{n-1} 57を第1の加算器54に供給する。第1の加算器54への入力58および60は、レジスタセル A_i およびアキュムレータセル C_i から得られる。第1の加算器54からの出力62は、モジュラスレジスタセル M_i からの入力64とともに、第2の加算器56の入力に接続されている。加算器54は出力 $62 = \text{入力}60 + (\text{入力}58 \text{ および 制御}57)$ という演算をパフォームする。この演算を図5(b)に詳細に示す。

ついで、第2の加算器56からの出力はアキュムレータセル C_i に接続されている。第2の加算制御信号66はアキュムレータ C 46の最上位のセル C_{N-1} から得られる。アキュムレータ C の最上位のビット C_{N-1} がセットされたとき、当然に、モジュラスベクトル m によるアキュムレータ C での部分積のモジュラ簡約が、第2の加算制御信号66により実装される。図5(c)に詳細に示すように、加算器56は、出力 $= \text{入力}62 + (\text{入力}64 \text{ および 制御}66)$ という演算を行う。Bレジスタはクロックシフトレジスタである。コントローラ20から供給することができるクロック信号CLK168は、部分積が計算される度に、このレジスタの内容を左にシフトさせる。

図6を説明する。図6は図5のビットスライス41の詳細な回路実装を示す。この回路実装は有限体乗算を行なうためのものであって、参照番号70で示す。図6のビットスライス i 、70を説明する。図6では、説明のために、ビットスライスは3つしか示していない。セル a_i は、ANDゲート72により、加算制御信号 b_{n-1} とAND演算される。ANDゲート72の出力74は、アキュムレータ C の隣接するセル C_{i-1} からの入力78とともに、XORゲート76の入

力に接続される。よって、項「 $c_{j-1} + b_j a_j$ 」の計算が実装される。項「 $c_{n-1} m_j$ 」は、ANDゲート84を利用して、信号 $c_n 80$ と $m_j 82$ をAND演算することにより、実装される。ANDゲート84の出力86は、XORゲート76の出力88とともに、XORゲート84の入力に接続される。XORゲート84

の出力90は、セル $C_j 92$ に接続される。よって、式「 $c_j = c_{j-1} + b_j a_j + c_{n-1} m_j$ 」が実装される。この汎用の逐次乗算器により、2つの n ビット有限体元の積が n クロックサイクルで生成されることになる。同期カウンタは、コントローラ20に含めることができるものであって、繰返し回数の制御を行うものが好ましい。以上の記述は、加算器54が整数加算器のビットスライスであって、加算器56が整数減算器のビットスライスであるときに、整数モジュラ乗算に適用される。このことは後程説明する。

加算

有限体 F_{2^n} 中の乗算に関連して、回路を説明したが、その他の計算オペレーションも容易にパフォームすることができる。有限体加算は桁上げが生じないので、この点で、整数算術より有利である。有限体サム(sum)の計算では、有限体中の2つの元 a および b の加算が、単に、 a と b のXORであるので、XORゲートを注目レジスタの各セルに導入するだけでよい。したがって、図5に戻ると、入力100はセル B_i から第1の加算器54に供給され、第2の加算器56は簡約に使用される。ついで、加算器54からの出力はセル C_j に直接書き込まれる。オペランドがレジスタ a および b に移動された後で、単一のクロックサイクルで、加算をパフォームすることができる。その加算をALUでパフォームするのは可能であり、その結果をレジスタファイルの汎用レジスタにライトバックするのも可能である。整数加算では、加算器54は整数加算器のビットスライスであり、整数加算結果に基づきモジュラオーバーフローか否かを検査しなければならない。この状態が生じた場合には、整数減算器のビットスライスである加算器56は、その結果を簡約するのに用いられる。

二乗

ある数を二乗するには、異なる2つの数の乗算と同じ時間でパフォームするこ

とができる。多項式基底における二乗は、特定の既約 (irreducible) が二乗展開と明示的に結線された (hardwired) 場合は、単一のクロックサイクルでパフォーマンスすることができる。あるいはまた、同じ入力を乗算して二乗をパフォーマンス

ことができる。

反転

F_{2^n} の有限体元の反転は、ユークリッドの互除法を使用してパフォーマンスことができ、また、追加のコントロールロジックを有する4つの特殊レジスタを利用してパフォーマンスすることができる。この反転は、シフトが加算と同時に行われる場合 (これは加算の出力を次のレジスタセルに結線することによって容易に実装される) には、 $2n$ サイクルで完了する。

この反転で使用されるレジスタは、A、B、M、およびCである。便宜上、これらのレジスタを概略的に図7に示す。MにはUL、CにはLL、AにはUR、BにはLRとラベル付けがしてある。再度、この反転を、ビットスライス110に関連して記述することができる。

反転のオペランドは、一般に、反転する元 g と、既約多項式 f またはモジュラス m (後述) と、ビットベクトル「0」と、ビットベクトル「1」である。ULレジスタ116には f または m がロードされる。LLレジスタ118には g がロードされ、URレジスタ112には「0」が、LRレジスタ114には「1」がロードされる。URレジスタ112およびLRレジスタ114では、セル UR_i および LR_i はXORゲート120でXOR演算されて、出力122が生じる。制御信号124は、可能な3つの入力のうち1つがセル UR_i および UL_i に書き込まれるかどうかを決定する。入力に隣接するセルまたは出力122からの左または右シフトである。制御信号Bは後程記載する状態表によって決定される。ULレジスタ116またはLLレジスタ118では、セル UL_i および LL_i はXORゲート126でXOR演算されて、出力128が生じる。制御信号130は、可能な2つの入力のうち1つがセル UL_i および LL_i に書き込まれるかどうかを決定する。入力に隣接するセル ($i-1$) または出力128からの左シフトであ

る。この場合も、制御信号130は後程記載する状態表によって決定される。

制御変数をULレジスタの長さ k_u と、LLレジスタの長さ k_l と仮定したとすると、 $\Delta = k_u - k_l$ となる。値 k_l および k_u は、好ましくは、同期ダウンカウンタで実装され、 Δ は好ましくは同期アップ／ダウンカウンタで実装される。カウ

ンタレジスタ k_u 、 k_l 、および Δ も用意されている。ULおよびLLレジスタは左シフトレジスタであり、URおよびLRレジスタは、ともに、左および右シフトレジスタである。

さらに、カウンタレジスタでは、 Δ には0がロードされ、 k_u はnに初期化される。制御ビットラッチは、「1」がアップカウントを示し、「0」がダウンカウントを示すトグル機能を有する。U/D制御は、最初、「1」にセットされる。この場合、ALUで反転を実行する制御装置に含まれるシーケンサは、次のような出力を有する。

deckl	デクリメント k_l	k_l
decku	デクリメント k_u	k_u
decDelta	デクリメント Δ	
incDelta	インクリメント Δ	
toggle	トグルアップ／ダウン	
lsUL	左シフト左上レジスタ	
lsLL	左シフト左下レジスタ	
lsUR	左シフト右上レジスタ	
lsLR	左シフト右下レジスタ	
rsUR	右シフト右上レジスタ	
rsLR	右シフト左下レジスタ	
outLR	出力右下レジスタ	
outUR	出力右上レジスタ	
dadd-lsLL	ダウンXORおよび左シフト左下レジスタ	
uadd-lsUL	アップXORおよび左シフト左上レジスタ	

インバータのオペレーションの概要を表す状態表は次のようになっており、M

u および C_l はそれぞれレジスタ U_L および L_L の上位ビットであり、 M_u および C_l は現在の状態を決定する。レジスタおよびカウンタ上でアクションがパフォーマンスされると、これによりインバータは新しい状態となる。このプロセスは、 k

u または k_l が0になるまで繰り返され、右レジスタ R_L または R_U の一方は g^{-1} を含み、もう一方はモジュラス自体を含むことになり、これは、後続の乗算または反転演算で使用するために、レジスタ m にリストア (restore) することができる。

U/D	k_u	k_l	Δ	M_u	C_l	アクション
X	0	X	X	X	X	OutLR
X	X	0	X	X	X	OutUR
1	$\bar{0}$	$\bar{0}$	0	0	1	Deck _u , dec Δ , lsUL, lsUR, toggle
1	$\bar{0}$	$\bar{0}$	$\bar{0}$	0	1	Deck _u , dec Δ , lsUL, rsLR
0	$\bar{0}$	$\bar{0}$	X	0	1	Deck _u , dec Δ , lsUL, lsUR
0	$\bar{0}$	$\bar{0}$	0	1	0	Deck _l , inc Δ , lsLL, lsLR, toggle
0	$\bar{0}$	$\bar{0}$	$\bar{0}$	1	0	Deck _l , inc Δ , lsLL, rsUR
1	$\bar{0}$	$\bar{0}$	X	1	0	Deck _l , inc Δ , lsLL, lsLR
0	$\bar{0}$	$\bar{0}$	0	1	1	Deck _l , inc Δ , Dadd-lsLL, lsLR, toggle
0	$\bar{0}$	$\bar{0}$	$\bar{0}$	1	1	Deck _l , inc Δ , Dadd-lsLL, rsUR
1	$\bar{0}$	$\bar{0}$	0	1	1	Deck _u , dec Δ , Uadd-lsUL, lsUR, toggle
1	$\bar{0}$	$\bar{0}$	$\bar{0}$	1	1	Deck _u , dec Δ , Uadd-lsUL, rsLR

整数算術

多項式表現と整数表現は非常に良く似ていることから、ALUでハードウェアを共有することが可能である。加算では、整数算術は、桁上げが必要であることから、複雑になるだけである。ALUの整数算術演算は、例えば乗算演算を利用すれば、最もよく説明することができる。

疑似コードで表した次の一連のステップを参照して、Zにおける乗算を説明す

る。前述したのと同様に、 a および b は乗算されるビットベクトルであり、 c は

a と b の積であり、 $c = (c_0, c_1, \dots, c_{n-1})$ である。

$C = 0$

$M = 0$

For i from 0 to $n-1$ do

$C_{ext} \leftarrow C$

For j from 0 to $n-1$ do

$C_j = (b_i (a_j) + m_j + c_j) \bmod 2$

$M^{j+1} = (b_j (a_j) + m_j + c_j) / 2$

ここで、

For i from 0 to $n-1$ do

$C_{cxt} \leftarrow C$

に対して、

$c_{j-1} = c_j$

$c_{j-1}^{ext} = c_j^{ext}$

となる。

同様に、このようにすれば、XORを減算器で置換し、しかも、 m レジスタに素数をロードした場合には、整数modulo p (integers modulo p) を反転させることができる。改善策であるが、桁上げ保存方法を採用することにより、桁上げ伝搬を遅らせることができる。

図6の実施形態で説明した有限体乗算の場合のビットスライス70を修正して、整数表現に対する乗算を含むようにすることができる、ことを観測することができる。注意すべきことであるが、整数乗算では、レジスタには、ビットベクトルが F_{2^m} とは逆順でロードされる、すなわち、レジスタの最左端のセルがビットベクトルのLSBを含む。整数乗算では、逐次 (successive) 部分積の間で、桁上げを実装する必要がある、さらに、部分積がモジュラスで簡約されていないので、逐次部分積の加算による桁上げを供給しなければならない。そこで、アキュムレータレジスタ C が拡張してあり、図5に示すように、新しいレジスタ $C_{ext} 49$ が設けてある。各部分積が形成される前に、アキュムレータ C の最下位ビット

ト（セル C_M ）を拡張レジスタ C_{ext} の最上位ビット（セル C_{ext_l} ）にシフトし、
 ついで、アキュムレータ C および C_{ext} は両方ともLSBに向けて1ビットだけ
 シフトされる。最終結果は C および C_{ext} で獲得され、 C_{ext} には、当該積の低位
 ビットが含まれる。このことは、上記オペレーション $C_{ext} \leftarrow C$ で表される。

図8を説明する。図8はビットスライス170を示す。ビットスライス170
 は図6のビットスライス70に類似している。したがって、同様のコンポーネン
 トは、識別するために、図6の説明で使した参照番号を100番台にして使用
 することにする。つまり、参照番号70は170となる。図8の編成が図6と異
 なる点は、モジュラスレジスタ m が桁上げレジスタとして使用され、モード選択
 信号 $Z/F_{2m}171$ が提供されるという、2つの重要な点である。

ここで、項 $c_j = c_{j-1} + b_j a_j + c_{n-1} m_j$ は、既に説明した有限体乗算でそう
 であったように、制御信号 b_m とレジスタセル A_j の内容との積で実装され、こ
 の積はANDゲート172で実装される。ANDゲート172の出力174はレ
 ジスタセル c_{j-1} の内容とXORゲート176によりXOR演算され、参照番号
 158で示す出力項 $c_{j-1} + b_j (a_j)$ が生成される。この出力信号は、AND
 ゲート160から得られた参照番号185で示す項 $c_{n-1} (m_j)$ と、XORゲー
 ト184を使用してXOR演算され、項 c_j が生成される。さらに、積 $b_j a_j$
 $, c_{j-1}$ 162と、積 $(c_{j-1} + b_j a_j, m_j)$ 163とのサム(sum)から、桁
 上げ項 m_j が生成され、セル m_j 182に書き込まれる。積の項162および16
 3はANDゲート164および166によってそれぞれ実装される。積の項16
 2と163のサムはORゲート167によって実装される。

モード選択信号 Z 171は、桁上げ入力信号 C_n 180とOR演算され、クロ
 ック信号169とAND演算168される。したがって、 $Z=0$ をセットするこ
 とにより、有限体算術が実装され、 $Z=1$ をセットすることにより、整数算術が
 実装される。

図8は、図6で既に説明した有限体乗算を、組合せ有限体／整数乗算器に変換
 するのに必要な修正を示す。乗算の低位のビットを集めるため、出力レジスタ C
 が拡張されることに留意されたい。 Z における計算はモジュラスなしでパフォー
 ムされるので、モジュラスレジスタ M は、部分積を簡約するためではなく、桁上

げのホルダとして使用される。制御信号 Z/F_2^{M171} は、ALUのための整数乗算回路をイネーブルにする。

最終桁上げ伝搬 (final carry propagation) は、マンチェスタリップルチェーン (Manchester ripple chain) によって提供することができ、レジスタ長が長いことから、1レイヤまたは2レイヤの桁上げスキップ機構によって拡張可能である。さらにnサイクルだけクロックすることも可能であり、桁上げ保存加算器が桁上げを完全にマージすることが可能である。

1つの入力はその入力において条件付きで補数をとることができ、しかも、加算器のLSBで「ホット」キャリインが行われる場合には、2の補数の減算は、桁上げ伝搬加算器で実装することができる。

乗算時のリップル桁上げは、桁上げスキップにより改良したとしても、許容でなくなるが、この桁上げ伝搬は、桁上げ保存加算器を使用すれば、ほぼ完全に除去することができる。このようにすると、部分積が冗長表現されるが、乗算が完了した後は解決される。

さらに別の実施形態では、ALU4は、図9に示すように、計算速度が線形に増加するように修正することができる。これは、特殊レジスタ16'からの連続ビットを一度に処理し、修正したサブALU190で示す追加回路を実装し、図9に示すようにインクリメント加算を処理することによって達成される。複数のビットを処理すると、速度が線形増加することになる。例えば、計算が順次にパフォームされる場合は、その順序中の2つ以上のステップを同時に実行することができる。この場合、コントローラ20'は特殊レジスタ16'からの2ビット以上の制御ビット194を処理することになり、制御装置の入力192は図9にマルチビットラインとして示す。

有限体に対して一度に2ビット実行する乗算器 (two-bit at a time multiplier) の回路図を図10に示す。この実装では、ビットスライス200はその数がXORゲート210の数の2倍であり、当該加算の2つの項を実装している。この乗算器は乗数から2ビットをとり、被乗数 a_j および a_{j-1} を2回だけ隣接してシフトすることにより加算し、モジュラス M_j および M_{j-1} を2回だけ隣接してシフトすることにより簡約する。このようにすると、モジュラス簡約 (modulus

reduction) で連続する2つの部分積が同時に生成され、したがって、全計算時間を半分にすることができるという効果がある。

特殊レジスタの上位(top)ビットがコントローラ20または20'用の制御ビットとして使用される、ことに留意されたい。このようにすると、オペランドがレジスタにロードされると、左揃えされ、したがって、制御が常に固定ビット位置から得られるという利点がある。しかし、その他のビット例えば下位(bottom)ビットを制御ビットとして使用することもできる。しかし、このようにすると、ハードウェアが複雑になることもある。

この場合も、Booth(または、修正Booth)記録などのオプションが可能となるので、マルチビット演算の計算速度がさらに線形的に増加する。

このようなALUは汎用レジスタに対して簡単な算術演算をパフォームする能力を有するものと仮定している。他の例のALUは全ての算術をALU内部レジスタに対してパフォームするものであり、汎用レジスタはこれらのレジスタとの間でリード(read)およびライト(write)のみを行う能力を有する。

このようなALUの機能には、リップル桁上げや、桁上げスキップ加算と桁上げ完了の組合せなど、何らかの桁上げ伝搬方法を利用した、整数加算が含まれる。

このようなALUは、有限体加算で使用される単純なXOR機能も提供する。整数および有限体表現(ビット順序)が逆であるので、体から整数への変換と、整数から体への変換に使用されるビット逆転(bit reversal)機構を設けると有利である。2つのシフトレジスタの頂部どうしを接続することにより、nクロックサイクルでこの機能が提供される。ここで、nは算術オペランドの長さである。

本明細書で与えた一般的なアーキテクチャは、ECとモジュラ指数算術との間でレジスタファイルを共用するだけでなく、共用制御レジスタに加えて、特殊レジスタおよび組み合わせロジックも共用する可能性がある。

以上、本発明の具体的な実施形態と具体的な用途について説明したが、種々の修正は、本発明の範囲を逸脱しない限り、当業者にとって可能である。例えば、

記載の実施形態では、特定のロジック回路について言及したが、例えば、ド・モルガンの法則を使用して等価な回路を使用することもでき、反転ロジック (inverted logic) が実装された場合には、相補形回路を使用することもできる、

ことに留意されたい。さらに、レジスタおよびビットベクトルのオリエンテーション、すなわち、左、右、上、下には、これらの方向の他の編成も含まれる。

本明細書で採用した項および式は、これらのものに限定されるものではなく、例として使用したものであり、これらの項および式を使用したことに、図示および記述した機構またはその一部分の均等物を排除する意図はなく、本発明の範囲内で種々の修正が可能であることを認識されたい。

【図1】

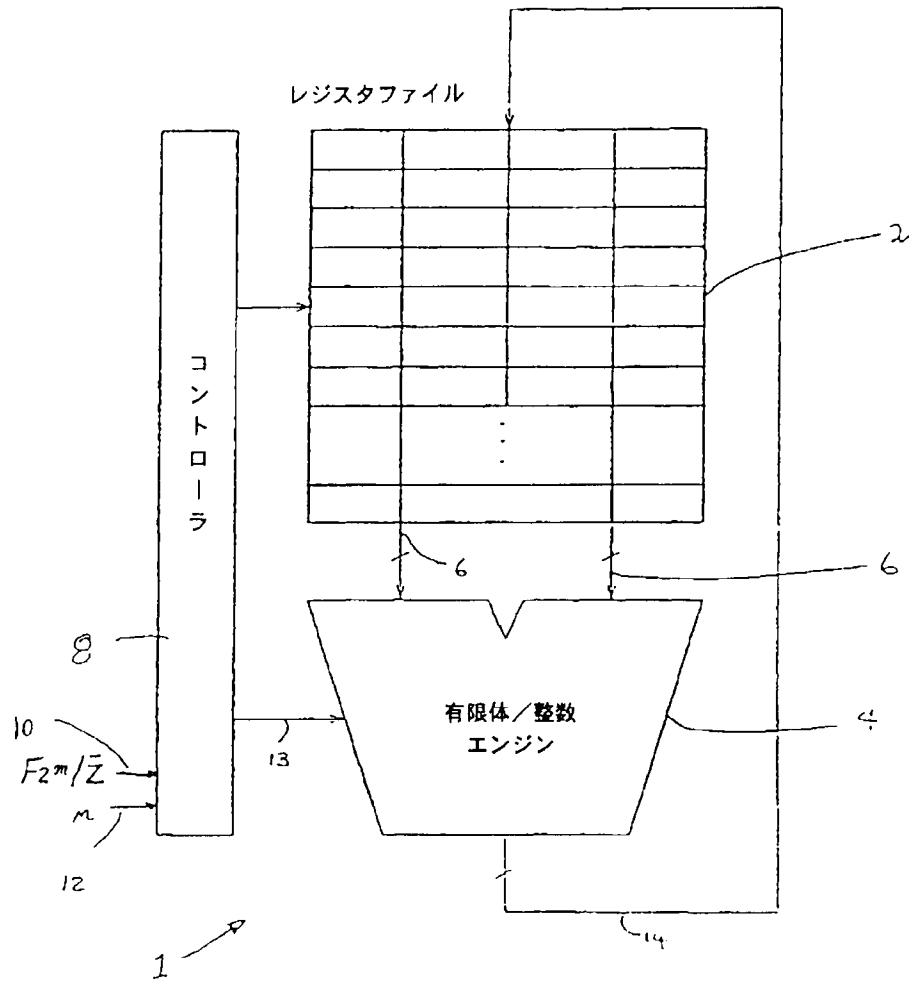


FIGURE 1

【図2】

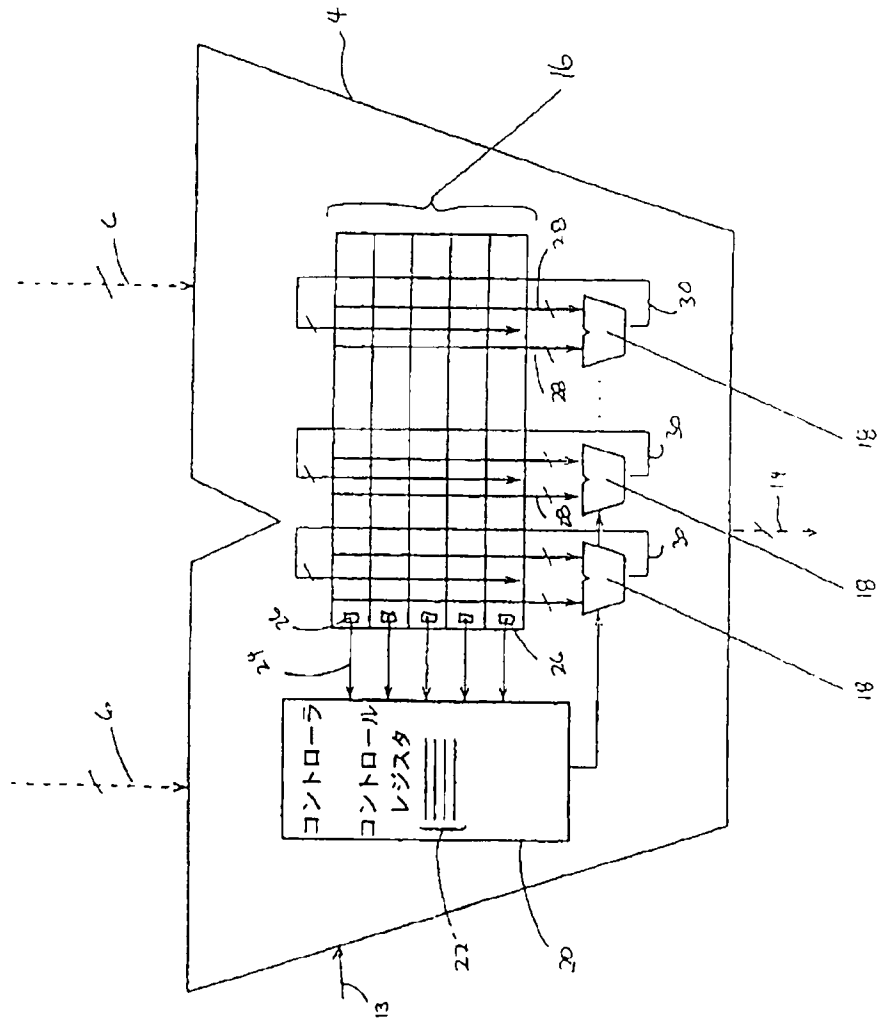


FIGURE 2

【図3】

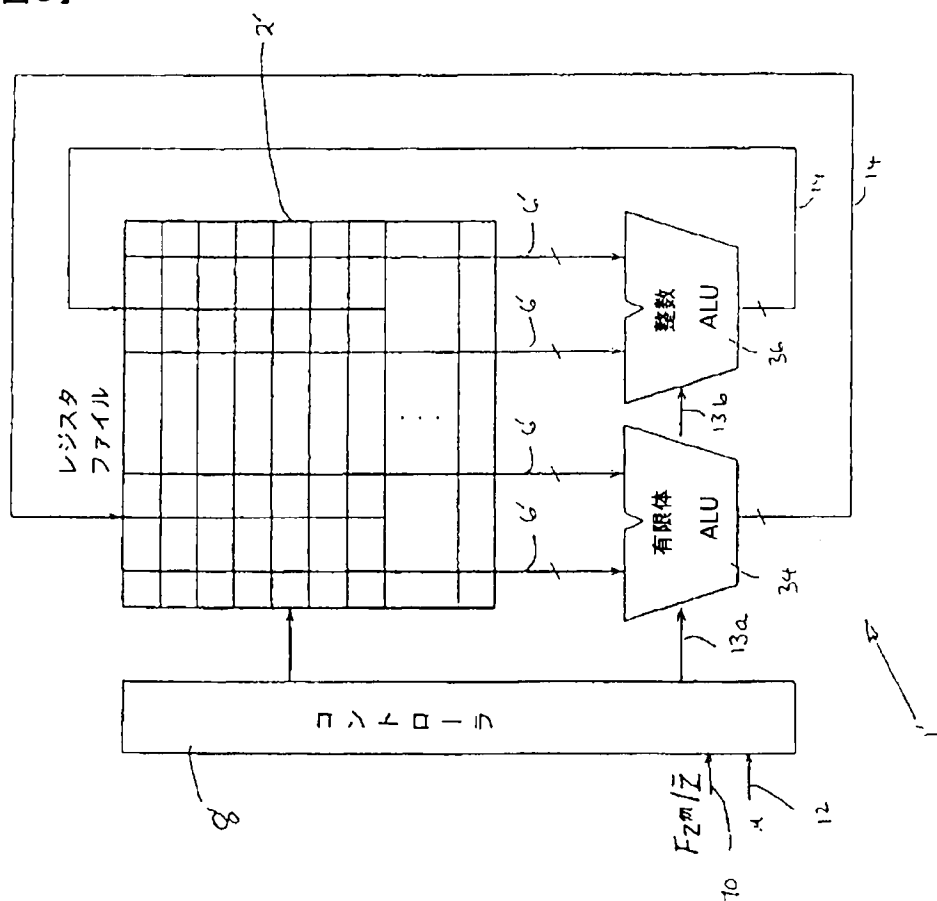


FIGURE 3

【図4】

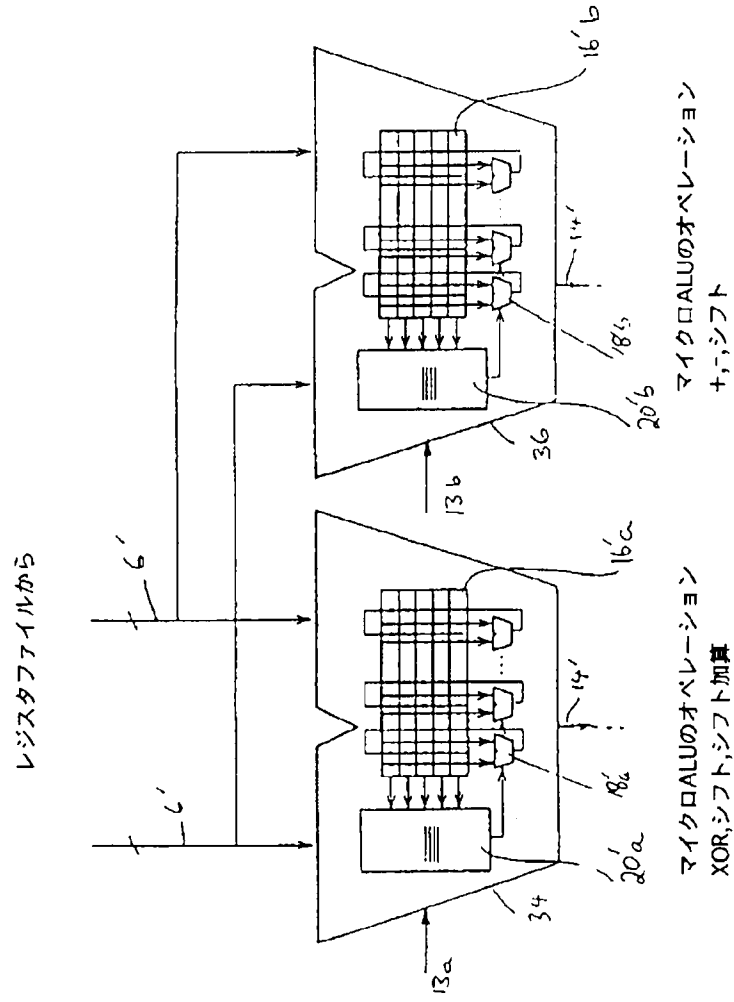


FIGURE 4

【図5】

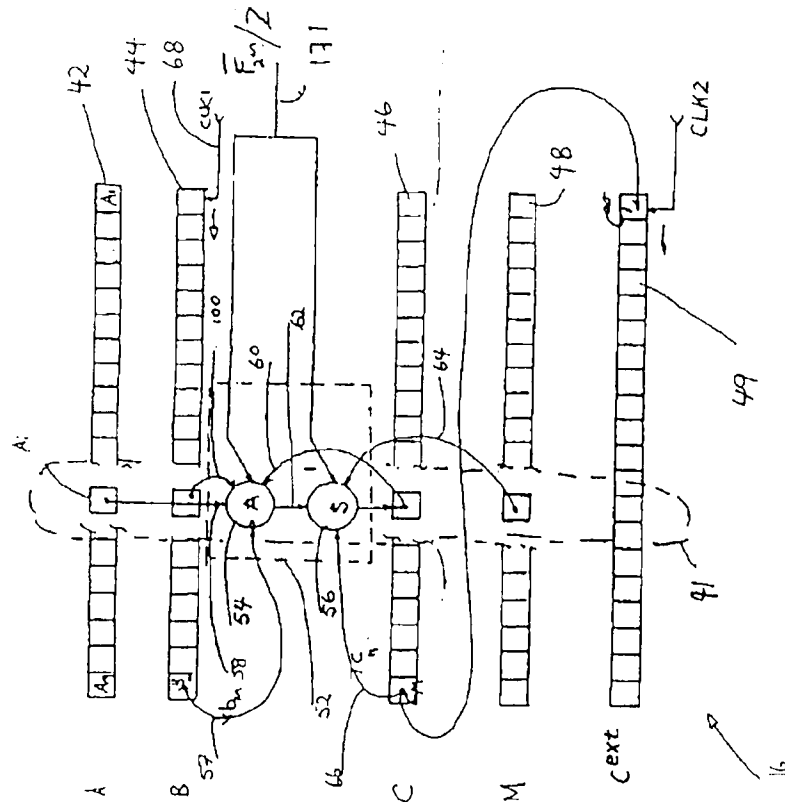
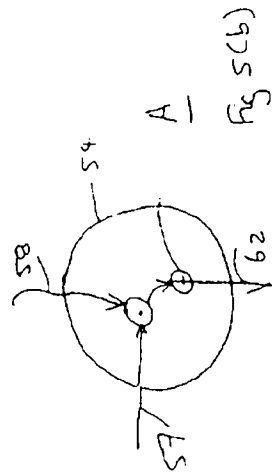
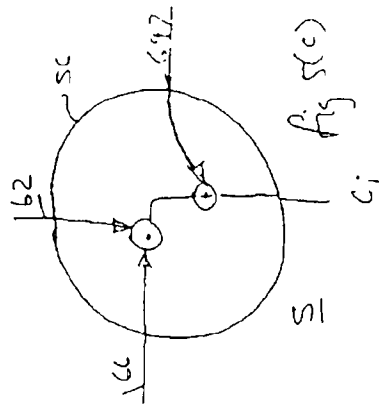


FIGURE 5(a)

【図 5】



【図6】

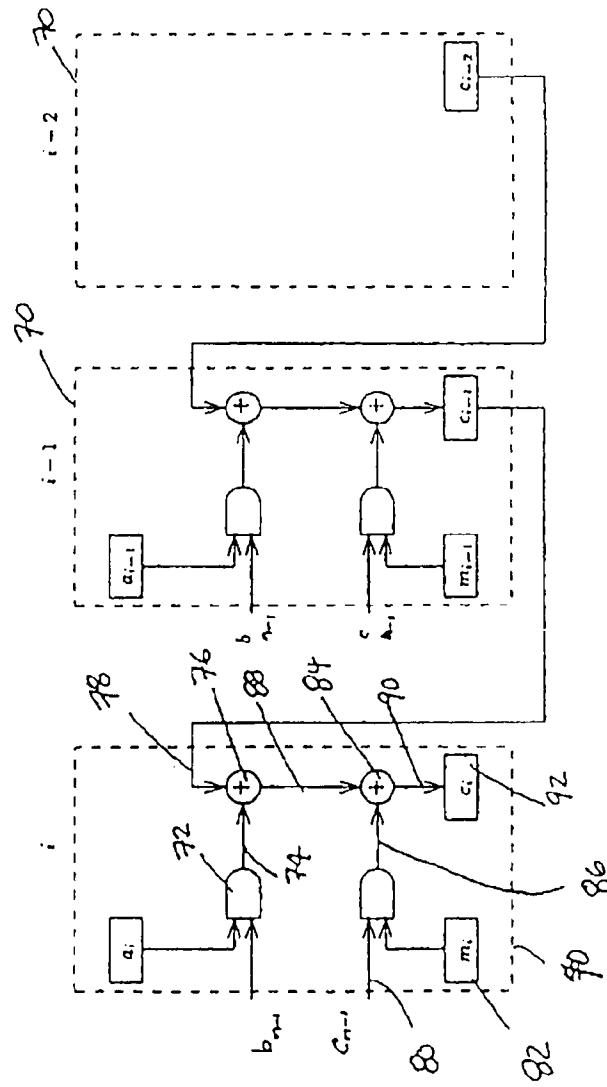


Figure 6

【図7】

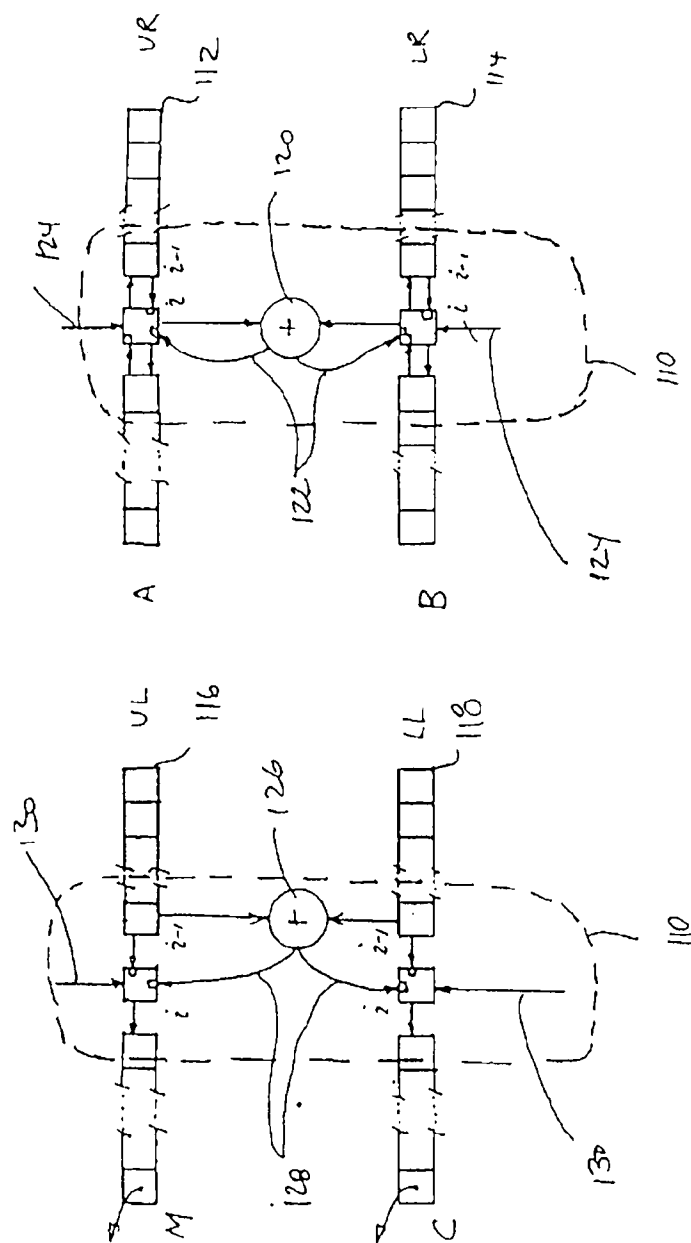


FIGURE 7

【図9】

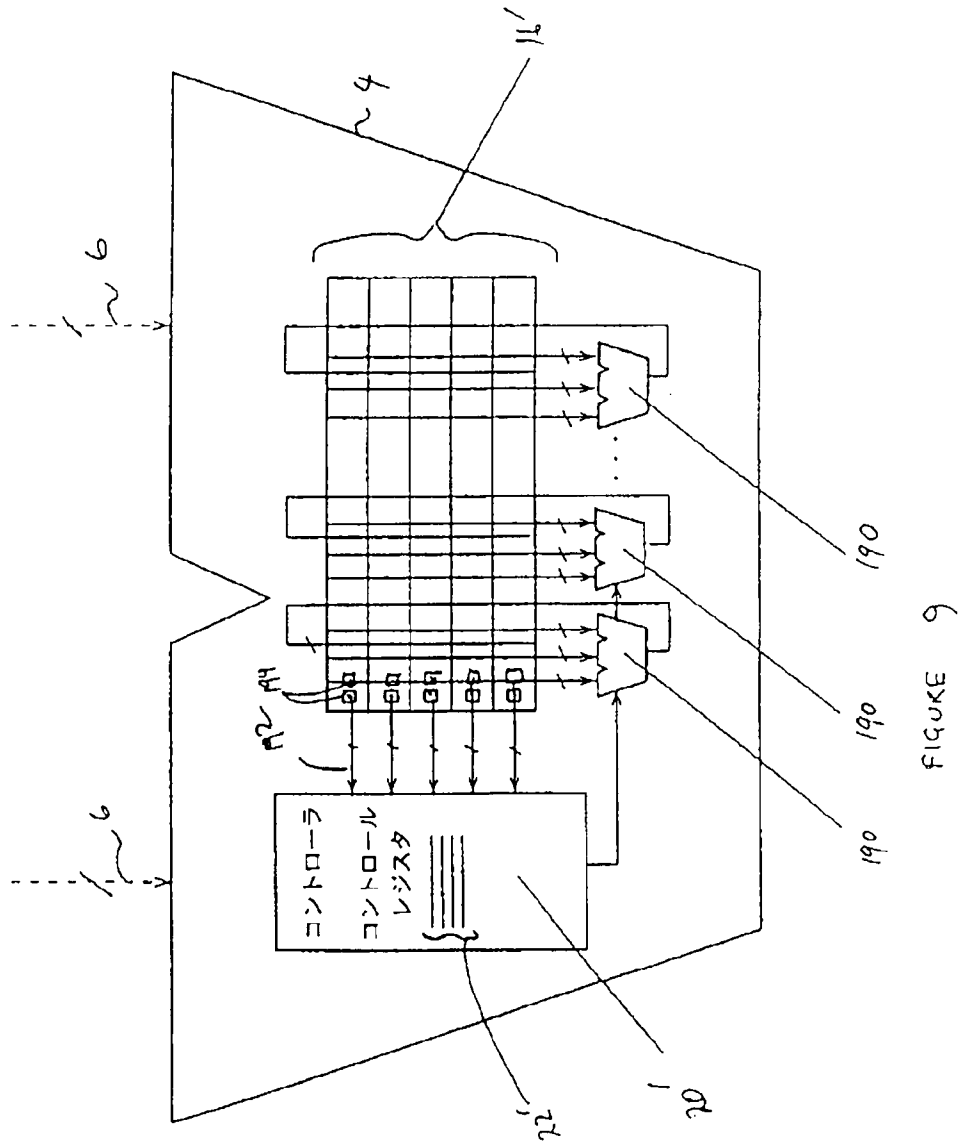
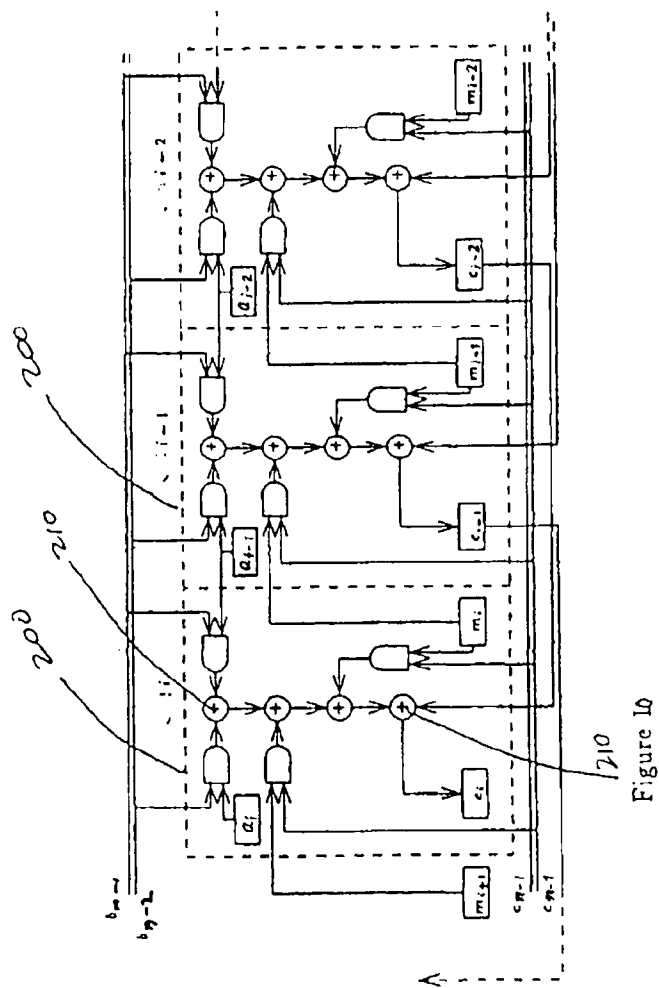


FIGURE 9

【図10】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER IPC 6 G06F7/72		International Application No. PCT/CA 98/00467
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 6 G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 268 854 A (IKUMI NOBUYUKI) 7 December 1993 see column 1, line 45 - column 2, line 8; figure 9	25
A	US 5 459 681 A (HARRISON CALVIN W ET AL) 17 October 1995 see column 1, line 37 - line 46	1,26
A	EP 0 267 836 A (THOMSON CSF) 18 May 1988 see figure 1	1,26
A	FUCHS K C: "CRYPTOGRAPHIC SIGNAL PROCESSOR" MOTOROLA TECHNICAL DEVELOPMENTS, vol. 27, 1 May 1996, page 81/82 XP000594556 see the whole document	1,26
<input type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "Z" document member of the same patent family		
Date of the actual completion of the international search 7 August 1998		Date of mailing of the international search report 13/08/1998
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel: (+31-70) 340-2040, Tx: 31 651 upo nl, Fax: (+31-70) 340-3016		Authorized officer Verhoof, P

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No.

PCT/CA 98/00467

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5268854 A	07-12-1993	JP 2054544 C	23-05-1996
		JP 4178785 A	25-06-1992
		JP 7085267 B	13-09-1995
US 5459681 A	17-10-1995	NONE	
EP 0267836 A	18-05-1988	FR 2605818 A	29-04-1988
		DE 3778649 A	04-06-1992
		US 4888778 A	19-12-1989

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW

(72)発明者 ガラント, ロバート.
カナダ エル7アール 2エヌ3 オンタ
リオ バーリントン パール ストリート
607—478

(72)発明者 ユリシク, アレクサンダー.
スロヴェニア共和国 1000 リュブリャナ
ビバプスカ 24エイ

(72)発明者 バデカール, アショク, ヴィ.
カナダ エル5アール 3ジー8 オンタ
リオ ミシソウガ コンスタレーション
ドライブ 2006—700